Each role on a Scrum-agile Team is vital to the success of a project. Starting with the Product Owner. When being the person who is talking to the client and responsible for taking what they are asking for and communicating that with the team it is an essential role that can make or break the outcome of a project. Although all members of the team have ownership of the whole project and not just their part, it really takes someone like the Product Owner to be able to step back and see the big picture. This specifically came into play when working on the SNHU Travel project with managing the backlog, When things changed in the process as they always do they were able to share that information with the team quickly and effectively so we were able to make adjustments to the program and show the new direction of wellness vacations that the client wanted.

“The Scrum Master is what is known in agile as a servant leader.”(Cobb, 37) There place on the team is to help facilitate whatever is needed to keep the team going. They assist the Product Owner in being more efficient and help facilitate making the team more efficient. The project could probably get done without a Scrum Master, but at the pace that agile sets with its sprints it would not be possible without someone looking to keep the ball rolling the whole time and anticipate and correct roadblocks that might come up in the process.

When dealing with the Developer role it is the job that most people think of when it comes to making a product. The thing is that this role can be done in so many different ways. When it comes to improving development being able to work with everyone and share what you need is so important to keep things moving and to move past roadblocks. When working on the SNHU Travel project, the developer was able to work on the project and share their time lines, as an example when adjusting the project for the clients new requests the developer was able to share that they did not think that this change would affect the projected time line, but if this was not the case and it would they would be able to share that information and either have more resources and people put on that part, or make other adjustments to the backlog to facilitate the needs of the developer.

The tester role in an agile team takes a much more involved roll then in a typical waterfall method. They are much more hands on with the development team throughout the whole process instead of just when the project has been built. By being involved throughout the whole process and the focus on having working parts that can be tested frequently and throughout the development process it makes it much more manageable and significantly easier to debug any issues that might come up as the code base that you are testing at any time is not overwhelming. In the SNHU project, we realized that the user stories had a lot of overlapping test requirements which allowed us to reuse test cases which in turn made testing easier, faster, and continually refined.

The three user stories that were focused on in the SNHU travel project were; A list of the best deals, The ability to filter by price, and personalized vacation options for repeat customers. All of these were able to be worked on to completion in each step to allow proper testing and to build off of what was previously worked on. By first getting a list for the hot deals, then working on the filtering options, and the finally working on the user specific list we were able to build off of the previous work and keeping moving forward. The personalized list was going to need filtering and the initial list setup created. Those other user stories would have already been tested, so the only thing that was needed to be focused on was how the logic of the personalization worked. By building in smaller complete sections and testing them it allowed a much more efficient development to testing to next step of the project flow.

When the project changed directions being in an Scrum-agile environment makes this a much more seamless process. Because the product owner is communicating with the client and then communicating with the team on a very regular basis the ability to make changes to what is being worked on is key. By using a backlog type system you are able to change what is the most important thing and cut down the backlog where needed, especially when changes have come in that might take a higher priority as something that client feels they need, which might push a want out of the list. By doing this and using things such as story points and burn down charts you should be able to fit in the new requirements and adjust what you are working on based on how difficult and how many story points this new addition has.

As an example of some communication of what is needed for our team in a collaborative manor I included a communication between myself as a tester answering questions from the developer on what the best way of us doing things to get the best results would be. This is but one example another one not included would be answering some questions from the project manager on how we want to track and manage where the project is at any point:

Hello Team,

I'm excited too transition from our old waterfall development cycle to the agile system. Ever since going through the course I am a firm believer in the agile process. I feel the most important component that the agile process brings is the Daily Scrum. Working in unison with everyone will ensure that we meet and exceed our customers expectations. Additionally, these smaller incremental Sprints should reduce our individual workload compared to the waterfall development.

I would like to know more from our Tester in how they will pass on critiques so that we can be efficient in fixing them. I also feel we should only do a retrospective at the end of our development cycle and not after each Sprint. I feel the agile process is considerably thorough and efficient. It would only slow down our moment on each Sprint. I would like to hear the teams feedback on this idea. Lastly, we should focus on creating dialogue; instead of a monologue. Remember we are a team, one entity. We are strong and weak together. Always show compassion with each other and strive to push ourselves to do our very best!

- Sergio Passos

Sergio,

As a tester I am super excited to be working on this next project together. In reference to your question I mentioned in my post, but will reiterate in yours since you asked. If you would be comfortable I would love to do some pair programming with you, I think it serves 3 benefits. One it helps me point out common things that fail tests that you might have missed, but we see time and time again. That way it can be adjusted before even writing tests. Secondly it helps us in testing get a better understanding of the code base, I think this is instrumental in testing and coming up with tests that we might not have ever thought of. Finally and probably the most important I think that by pair programming we can learn from each other, I can continue to learn what developers are looking from us in good tests, and good feedback, and developers can learn what testers are looking for in code. By working together it really focuses on the dialogue comment that you mentioned and the teamwork. I do also think this will help us to keep sprints moving instead of the development waiting for testing and vice versa.

- John Cooper

I think the biggest tool in the agile methodology is Scrum events specifically the daily scrum meetings. By being open about what you are working on, what has been completed, what new has come up that needs to be done, and most importantly what you are struggling with / need help with you are able to work through a project as a team instead of on your own. This allows you to put more time into what you are best at and allow others who might be better at something else that you are struggling with to help so you can keep moving forward.

I think that the SNHU Travel project went much more efficiently because of using the agile framework. This was most definitely the best approach. I believe we were able to provide a project that met the customers needs and desires while at the same time keeping a healthy and effective working environment. When it comes to development I don’t think there is a better system available at this time. Another thing that made this possible is the size of the team. I do not think that the Scrum-agile approach would be best if we were a team of 2 or a solo developer. I truly believe that the collaboration and teamwork of the multiple roles helps to shape the process into an enjoyable experience. This is harder in the smaller teams as you have less people to lean on for support in what you are working on. I also think if projects were not worked on in small teams and it was a huge department trying to coordinate in this manor it would be too much and a different approach might be better.